

擬似乱数検証ツールの調査開発

丹羽朗人<akito.niwa@toshiba.co.jp>, 小池正修<masanobu2.koike@toshiba.co.jp>,
柄窪孝也<kouya.tochikubo@toshiba.co.jp>, 福島茂之<shige@sitc.toshiba.co.jp>,
熊給英洋<hidehiro.kumakyuu@toshiba.co.jp>, 小阪和宏<kazuhiro3.kosaka@toshiba.co.jp>,
松岡賢<masashi2.matsuoka@toshiba.co.jp>

株式会社 東芝 e-ソリューション社 SI 技術開発センター

電子政府の実現に向けて、すべての申請・届出手続きを電子化するためには、例えば電子申請等における利用者認証のための電子署名やデータ保護のための暗号が必要となる。一般に暗号や電子署名のアルゴリズムにおいては、擬似乱数が広く用いられるため、擬似乱数技術の実装が必要となると考えられる。

そこで本調査開発では、乱数に関する文献及び評価ツールを調査し、擬似乱数の評価に有効と思われる検定とそうでないものとを分類する。また各検定法の関連を理論的に解析し、乱数を検定する上で必要最低限の検定法のミニマムセットを求め、求めたミニマムセットを実装した乱数検定ツールを開発することを目的とする。

1. はじめに

電子政府構想の基盤技術である情報セキュリティ技術では、鍵生成やチャレンジレスポンス認証など様々な場面で乱数を利用する。このため、システムの安全性評価には、システムで利用する乱数の評価が不可欠である。乱数の性質を調べるためには、統計的な手法を使ってその性質を解析するという手段が一般的であり、様々な統計的検定法が提案されている。しかしながら、提案されている検定方法の数はきわめて多く、また、乱数検定ツールや文献によって、採用している検定の種類や数はまったく異なっているのが現状である。

そこで本調査開発では、乱数に関する文献及び評価ツールを調査し、擬似乱数の評価に有効と思われる検定とそうでないものとを分類する。また各検定法の関連を理論的に解析し、乱数を検定する上で必要な検定法のミニマムセットを求める。そして、調査・分析結果に基づき定めたミニマムセットを実装した乱数検定ツールを開発することを目的とする。

開発した乱数検定ツールでは、調査で導出したミニマムセットの検定を実行する IPA 推奨乱数検定機能の他に、FIPS 140-2 [2,3] で定められた乱数検定を実行する FIPS 140-2 乱数検定機能、NIST Special Publication 800-22 [5,6] と同様の検定を行う NIST Special Publication 800-22 乱数検定機能などがある。

ニマムセットの検定を実行する IPA 推奨乱数検定機能の他に、FIPS 140-2 [2,3] で定められた乱数検定を実行する FIPS 140-2 乱数検定機能、NIST Special Publication 800-22 [5,6] と同様の検定を行う NIST Special Publication 800-22 乱数検定機能などがある。

2. 調査開発の目標と内容

2. 1 調査開発の目標

現在提案されている検定方法の数はきわめて多く、また、乱数検定ツールや文献によって、採用している検定の種類や数はまったく異なっているのが現状である。そこで本調査開発では、既存の乱数検定法や乱数検定ツールを調査し、数学的な考察や実際に乱数生成アルゴリズムからの出力に対して検定を行うことにより有効な検定とそうでないものとを分類し、乱数列を検定する上で必要最小限のもので構成されるミニマムセットを導出し、さらに、導出されたミニマムセットの検定を実行する乱数検定ツールを開発することを目標とする。

2. 2 調査開発の内容

2. 2. 1 調査の概要

(1) 調査対象

本調査開発では以下の5つの文献、ツールを中心に調査を行った。

A) The Art of Computer programming, Seminumerical Algorithms [1]

12種の検定が示されており、検定法は、区間 $[0,1)$ 上を分布する乱数列(小数)に対する検定と0からd-1の間で分布する乱数列(整数)に対する検定の2つに分類される。

B) FIPS 140-2 [2, 3]

4種類の検定法を採用している。0と1からなる20000ビットの乱数列だけが検定対象である。良い乱数かどうかは、検定結果が予め定められた範囲に含まれているかどうかで判断する。

C) 乱数 [4]

7種類の検定が示されている。文献[1]同様検定法は、区間 $[0,1)$ 上を分布する乱数列(小数)に対する検定と0からd-1の間で分布する乱数列(整数)に対する検定の2つに分類される。

D) A Statistical Test Suite For Random and Pseudorandom Number Generators for Cryptographic Applications [5, 6]

このツールで採用されているすべての検定は、0と1からなる乱数列を対象としている。また、このツールでは、検定ごとにp-valueが得られる。p-valueとは、真の乱数生成器が検定を行っている系列よりも乱数らしからぬ系列を生成する確率である。このツールでは、1本の標本系列に対する仮説検定で乱数生成アルゴリズムを評価するのは無意味であるという考え方から、複数の標本系列(NISTでは1000程度を推奨している)

に対し検定を行い、得られたp-valueの分布から乱数列を評価している。

E) DIEHARD [7]

DIEHARDで採用されているすべての検定は、0と1からなる乱数列を対象としており、乱数列の長さは、10MByteから11MByte程度必要である。DIEHARDでは、18種類の検定から200以上のp-valueが得られる。検定対象が良い乱数列の場合には、得られたp-valueは区間 $[0,1)$ 上に一様に分布し、また、良くない乱数列の場合には、p-valueの分布に偏りが生じる。なお、DIEHARDでは、多数のp-valueが得られるだけで、検定対象を良い乱数と判断するための基準は述べられていない。

(2) 数学的根拠の明確化とミニマムセットの導出

次に、本調査開発ではA)–E)で採用されている検定法を

- ブロック単位の頻度に関する検定
- パターンの出現に関する検定
- 状態遷移・ランダムウォークに関する検定
- 一様性・圧縮可能性に関する検定
- 周期性に関する検定
- その他の検定

の6つに分類し、それぞれの検定の目的および数学的根拠を明確にし、乱数列を検定する上で必要最小限のもので構成されるミニマムセットを導出した。なお、根拠が明確でない検定や0と1からなる乱数列に対しては意味がない検定等は、ミニマムセットから除外している。

本調査開発で得られたミニマムセットは以下の通りである。

- ブロック単位の頻度に関する検定
 - 高次元度数検定
 - ブロック単位の度数検定
 - ブロック単位の最長連検定
 - 8ビット中の文字数検定
 - 特定位置の8ビット中の文字数検定

- OPERM5 検定
- パターンの出現に関する検定
 - ビット列検定
 - OPSO 検定
 - OQSO 検定
 - バースデイ空間検定
- 状態遷移・ランダムウォークに関する検定
 - 累積和検定
- 一様性・圧縮可能性に関する検定
 - 系列検定
- 周期性に関する検定
 - 線形複雑度検定

なお、高次元度数検定は A)–E) で採用されている検定法ではなく、本調査開発で新たに導入した検定法である。

(3) 統計的な乱数評価の不十分性

本調査開発では、NIST のツールや DIEHARD に付属する乱数生成アルゴリズムからの出力を用いて導出したミニマムセットの有効性を検証した。しかしながら、漸化式

$$X_i = aX_{i-1} + c \bmod M$$

で表現される線形合同法(Linear Congruential)からの出力などは、導出したミニマムセットのすべての検定をパスしてしまう。一般に、シミュレーションなどに使う場合、線形合同法からの出力は乱数として十分な性質を持っているとみなすことができるが、暗号に利用する場合は、安全でないことが知られている[12-18]。したがって、統計的な乱数検定だけでは暗号で用いる乱数の評価として不十分なこと分かる。この結果から、暗号で用いる乱数列を評価する場合は、生成アルゴリズムに出力が一様でない、あるいは、周期が短い等の欠点があるものは初めに除外し、欠点が見つからないアルゴリズムからの出力に対し統計的な検定を行いその性質を調べるといった、理論的な評価と統計的な評価を両方行う必要があることが分かる。

2. 2. 2 開発ツールの概要

(1) 検定機能概要

本開発ツールは、大きく次の 4 つの検定機能を持っている。

(a) FIPS140-2 乱数検定機能

2. 2. 1 B) で述べた FIPS140-2 に規定されている 4 種類の検定法による乱数検定を行う場合に使用する。

(b) SP800-22 乱数検定機能

ここで言っている SP800-22(Special Publication 800-22) とは 2. 2. 1 D) で述べたツールのことを指している。このツールで採用されている 16 種類の検定法による乱数検定を行う場合に使用する。

(c) IPA 推奨乱数検定機能

2. 2. 1 (2) の調査結果により求めたミニマムセットによる乱数検定を行う場合に使用する。本機能の使用のみで十分信頼できる検定結果を得ることができる。

ツールでは、IPA 殿がこの乱数検定機能を推奨するであろうと考えて、「IPA 推奨乱数検定機能」という名称を付けているが、実際にはこの名称は、乱数検定に必要なミニマムセットであることを意味しているのみであることを注意する。

(d) 個別検定機能

2. 2. 1 の調査結果から得られたすべての有効な検定法 28 種類の中から個別に選択し、乱数検定を行う場合に使用する。

(2) 開発ツールの構成

本開発ツールは、大きく GUI とライブラリから構成される。

(a) GUI 機能

グラフィカルユーザインターフェース (GUI) を介して、(1) の各検定機能の選択、および各検定機能を実行するために必要なパラメータ (乱数列、ファイル名など) の入力を行う。

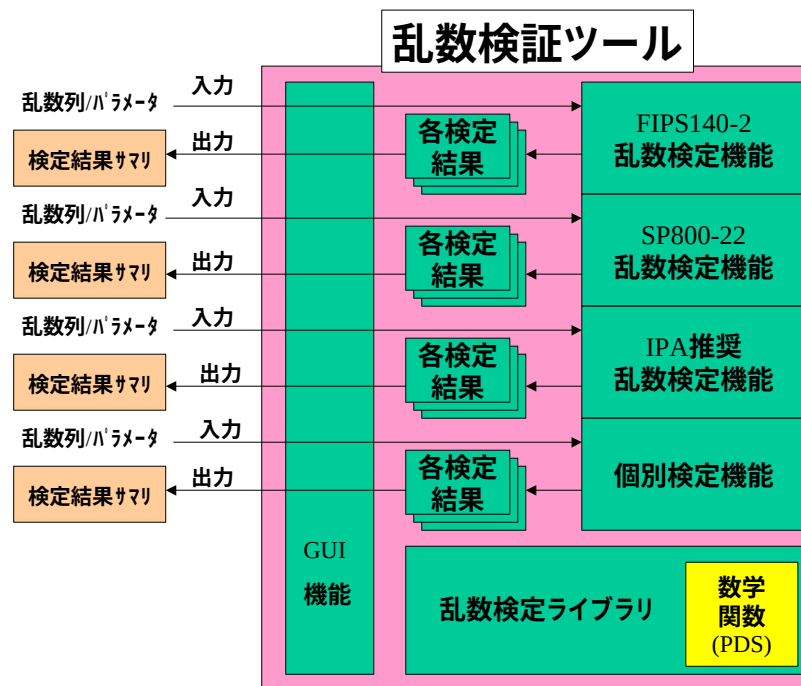


図1. 乱数検定ツールの機能関連図

乱数検定ライブラリの中から、選択された各検定機能に含まれる各検定法を呼び出して実行し、各検定結果及び検定結果サマリを出力する。

(b) 乱数検定ライブラリ

(1)の各検定機能で用いられる検定法の集合である。各検定機能を実行する際に、本ライブラリ内から対象となる検定法が呼び出される。

また、本ライブラリ内には、パブリックドメインソフト（PDS）の数学関数を組み込む。

以上の構成を図1に示す。

(3) 開発ツールの操作

ここでは、開発ツールの実際の操作画面を見ることによって検定の流れを紹介する。作業は①～⑥の手順で行われる。

① 初期画面（図2）

乱数検定ツールを起動するとこの画面が現れる。

4つの検定機能から実行する機能を選択できる。

② 乱数検定機能画面（図3）

この画面は①で SP800-22 を選択したときの画面である。ここでこの検定種類に属する検定機能の追加・削除を行うことができる。

③ 共通パラメータ設定画面（図4）

②で「共通パラメータ設定」ボタン押下により現れる。検定対象乱数列の情報を入力する。

④ 検定パラメータ設定画面（図5）

②で各検定のパラメータ設定を行うことができる。デフォルトで規定値が入っている。

④ 検定結果画面（図6）

③④の設定後、②で図3の「実行」ボタン押下により検定が実行され、検定結果サマリがこの画面で表示される。

⑤ 詳細結果画面（図7）

④で図6の「詳細表示」ボタンを押下することにより、各検定の p-value の分布等詳細情報を得ることができる。

⑥ 検定ログ画面（図8）

④の図6で、ログを表示させたい検定法を選択して「検定ログ表示」ボタンを押下することにより、選択した検定法のログを表示させることができる。

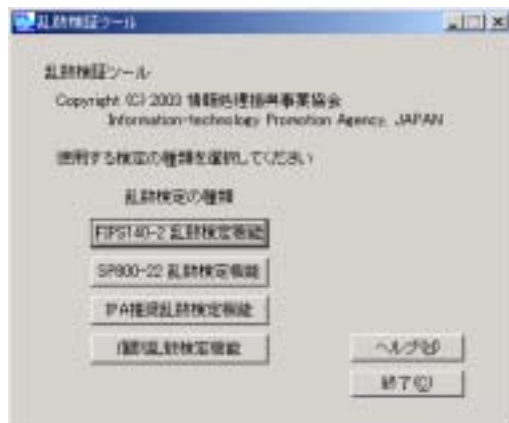


図 2. 初期画面



図 6. 検定結果画面

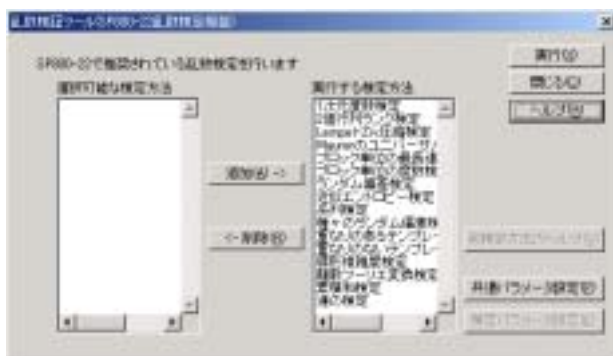


図 3. 乱数検定機能画面



図 7. 詳細結果画面



図 4. 共通パラメータ設定画面



図 8. 検定ログ画面



図 5. 検定パラメータ設定画面

3. 本年度の活動状況

本年度は、既存の乱数検定法や乱数検定ツールを調査し、数学的な考察や実際に乱数生成アルゴリズムからの出力に対して検定を行うことにより、有効な検定とそうでないものとを分類し乱数列を検定する上で必要最小限のもので構成されるミニマムセットを導出した。さらに、導出されたミニマムセットの検定を実行する乱数検定ツールを開発した。

4. 外部発表及び成果物

(1) 外部発表

2003 年 3 月 7 日現在で外部発表は特にない。

(2) 成果物

本調査開発テーマの成果物として納入した物件は以下の通りである。

- 成果報告書(調査報告書、開発報告書)
- 乱数検証ツール (バイナリ)
- 乱数検証ツール (ソースコード)
- 乱数検定ライブラリ (バイナリ)
- 乱数検定ライブラリ (ソースコード)
- 外部設計仕様書
- 試験仕様書
- 試験成績書
- 取扱説明書

5. 今後の課題

2. 2. 1(3)で述べた通り、暗号で用いる乱数列を評価する場合は、統計的な評価だけでは不十分であり、乱数生成アルゴリズムの理論的な評価も同時に行う必要がある。今後の課題としては、統計的な検定法だけでも十分に乱数の評価が可能な乱数検定法の開発などが挙げられる。

6. まとめ

本調査開発では、既存の乱数検定法や乱数検定ツールを調査し、数学的な考察や実際に乱数生成アルゴリズムからの出力に対して検定を行うことにより、有効な検定とそうでないものを分類し乱数列を検定する上で必要最小限のもので構成されるミニマムセットを導出し、さらに、導出されたミニマムセットの検定を実行する乱数検定ツールを開発した。

7. 参考文献

[1] D. Knuth, The art of computer programming,

seminumerical algorithms, Vol.3.

[2] NIST, FIPS PUB 140-2, ``Security requirements for cryptographic modules,"

[3] NIST, FIPS PUB 140-2, ``Change Notice 2," 2002.

[4] 伏見 正則, ``乱数," 東京大学出版.

[5] NIST, Special Publication 800-22, ``A Statistical Test Suite For Random and Pseudorandom Number Generators for Cryptographic Applications," 2001

[6] NIST, Special Publication 800-22, ``NIST Statistical Suite,"

[7] G. Marsaglia, ``DIEHARD," <<http://stat.fsu.edu/~geo/diehard.html> or <http://stat.fsu.edu/pub/diehard/>>

[8] P. Hellekalek, ``Tests for random numbers," <<http://random.mat.sbg.ac.at/tests/>>

[9] J. Walker, ``A Pseudorandom Number Sequence Test Program," <<http://www.fourmilab.ch/random/>>

[10] The Information Security Research Centre at Queensland University of Technology, ``Crypt-X," <<http://www.isrc.qut.edu.au/resource/cryptx/>>

[11] G. Louchard and W. Szpankowski, ``On the average redundancy rate of the Lempel-Ziv code," IEEE Trans. on Inform. Theory, Vol. 43, No. 1, 1997.

[12] J. A. Reeds, ``Cracking multiplicative congruential encryption algorithm," in Information Linkage Between Applied Mathematics and Industry, P.C.C. Wang, ed., Academic Press, pp.467-472, 1979.

[13] J. A. Reeds, ``Solution of challenge cipher," Cryptologia, Vol. 3, No. 2, pp.83-95, 1979.

[14] J. B. Plumstead, ``Inferring a sequence generated by a linear congruence," Proceedings of the 23rd IEEE Symposium on the Foundations of Computer Science, pp.153-159, 1982.

- [15]J.C. Lagarias and J. Reeds, ``Unique extrapolation of polynomial recurrences," SIAM Journal on Computing, Vol. 17, No. 2, pp.342-362, 1988.
- [16]H. Krawczyk, ``How to predict congruential generators," Advances in Cryptology CRYPTO '89, pp.138-153, 1990.
- [17]A.M. Frieze, J. Hastad, R. Kannan, J. C. Lagarias and A. Shamir, ``Reconstructing truncated integer variables satisfying linear congruences," SIAM Journal on Computing, Vol. 17, No. 2, pp.262-280, 1988.
- [18]J. Stern, ``Secret linear congruential generators are not cryptographically Secure," Proceedings of the 28th Symposium on the Foundations of Computer Science, pp.421-426, 1987.